

This chapter describes the MS-DOS commands that are not specifically dedicated to batch processing files. The commands are in alphabetical order.

This chapter covers:

- **Command Syntax**
- **MS-DOS Commands**

## Command Syntax

---

The following notation indicates how you should enter MS-DOS commands. Just like the rules of English grammar and usage, MS-DOS commands must be entered in an exact form — called command syntax — in order to have the correct action happen.

The following rules of command entry are used throughout this chapter to explain the proper way to express an MS-DOS command and all of its optional modifiers.

- You must enter any words shown in CAPITAL LETTERS. These words must be entered exactly as shown. MS-DOS accepts either upper or lowercase entries, but not misspellings.
- Anything enclosed in angle brackets < > must be entered. You supply the text. For example, enter the name of your file when <filename> is asked for in a command. Don't type the angle brackets. This would cause a problem.
- Items in square brackets [ ] are optional. Enter only the information asked for within the brackets.

Don't type the square brackets. This would cause a problem.

- Items in braces { } indicate that you must make a choice between two entries. You must enter one or the other unless those choices are enclosed in surrounding square brackets.
- An ellipsis (...) indicates that you may repeat an entry as many times as you need.
- Enter all punctuation shown (with the exception of angle and square brackets, ellipses, braces, or in special cases, vertical bars), such as commas, equal signs, question marks, colons, or slashes. These punctuation marks are essential and must be entered exactly as shown.
- A vertical bar (|) appearing in a command entry means one of two things. In one case, the | indicates a toggle switch. Where it appears one or the other choice can be made. It means "this" or "that" choice must be made. See the BREAK command for an example. In this case, the vertical bar is not typed in the command line.

In another case the vertical bar is used in the command entry to pass the results from one command to the next command on the command line. When | is used in this way, the results from the first command are passed to the second MS-DOS command for further processing. The second MS-DOS command is a "filter." The | symbol is called a "pipe."

See the SORT command for an example of piping the output from one command to another.

**DIR | SORT >DIREC.FIL**

sends the results of a DIR command to the SORT command, then writes the SORT results into a file named DIREC.FIL. Note the use of the > symbol.

MS-DOS normally assumes that input comes from your keyboard and that output is displayed on your screen. You can redirect the flow of input and output using the < and > characters. Input can come from a file instead of the keyboard, and output can go to a file instead of the screen. The pipe character can also be used with redirection characters to make complex MS-DOS operations easier to do.

For example,

**DIR**

displays a directory listing on your screen.

If you enter

**DIR >DIRFILE**

the output of the DIR command is written into a file on the default drive named DIRFILE. If the file named DIRFILE is not already on the default drive, it is created. If it already exists, the output from the DIR command writes over the contents of the existing file.

# MS-DOS Commands

---

The following MS-DOS commands are described in this chapter.

Note that synonyms for commands are enclosed in parentheses.

ASSIGN	Routes requests for one drive to another
BACKUP	Backs up files from a fixed disk
BREAK	Sets CONTROL-C check
CHDIR	Changes directories; prints working directory (CD)
CHKDSK	Scans the directory of the default or designated drive and checks for consistency
CLS	Clears screen
COMP	Compares files
COPY	Copies file(s) specified
CTTY	Changes console TTY
DATE	Displays and sets date
DEL	Deletes file(s) specified (ERASE)
DIR	Lists requested directory entries
DISKCOMP	Compares the contents of two diskettes
DISKCOPY	Copies disks
ECHO	On/off command for batch file processing echo feature
ERASE	Same as DELete
EXE2BIN	Converts executable files to binary format

EXIT	Exits command and returns to lower level
FC	Compares files
FDISK	Sets fixed disk partitions
FIND	Searches for a constant string of text
FOR	batch command for iterative processing of MS-DOS commands
FORMAT	Formats a disk to receive MS-DOS files
GOTO	Batch command directing jump to a specified location
GRAPHICS	Prints graphics from the screen on a graphics printer
IF	Batch command conditional
MKDIR	Makes a directory (MD)
MODE	Sets display, communications, and serial printer environments
MORE	Displays output one screen at a time
PATH	Sets a command search path
PAUSE	Batch command wait for input from keyboard
PRINT	Background print feature
PROMPT	Designates command prompt
RECOVER	Recovers a bad disk
REM	Displays a comment in a batch file
REN	Renames first file as second file (RENAME)
RESTORE	Restores files
RMDIR	Removes a directory (RD)
SET	Sets one string value to another
SHIFT	Batch command that increases the

---

	number of replaceable parameters
<b>SORT</b>	Sorts data alphabetically, forward or backward
<b>SYS</b>	Transfers MS-DOS system files from drive A: to the drive specified
<b>TIME</b>	Displays and sets time
<b>TREE</b>	Displays directories and their contents
<b>TYPE</b>	Displays the contents of file specified
<b>VER</b>	Prints MS-DOS version number
<b>VERIFY</b>	Verifies writes to disk
<b>VOL</b>	Prints volume identification number

# ASSIGN

---

**Purpose** The ASSIGN command tells MS-DOS to direct all requests from one drive to another drive.

**Syntax** ASSIGN [drivespec1 = drivespec2]

**Comments** Enter only the drive letters. Do not enter the colons on the command line.

Drivespec1 is source drive, drivespec2 is the new destination drive.

Entering the ASSIGN command without any original or destination parameters resets the current assignments to the default values.

ASSIGN could be useful in an AUTOEXEC.BAT file.

ASSIGN works unpredictably when used with the PRINT command. Do not use in this combination.

DISKCOPY and DISKCOMP commands ignore drive assignments.

---

**Purpose** The BACKUP command makes backup copies of one or more files from a fixed disk onto a diskette.

**Syntax** BACKUP [<d:>][<path>][<filespec>]  
d: [/S][/M][/A][/P][/D:<date>]  
[/T:<time>][/L:fname]

**Comments** Use this command when you want to recover unused space on your fixed disk drive or are going to create new partitions on the fixed disk.

The first parameter you enter is the fixed disk file(s) to back up.

The second parameter is the backup disk drive.

Be careful. Unless otherwise specified, the copies of the old files on the backup diskette are erased before the new files are written on it.

This backup program is compatible with those supplied by other manufacturers, except that the /P and /T options may cause incompatibilities

## BACKUP

---

between files backed up with different versions of the backup program.

The BACKUP command sets exit codes that can be used in a batch processing file with an IF command. These exit codes are:

- 0     normal completion
- 1     no files found
- 2     BACKUP stopped by user
- 3     BACKUP stopped by error

Files duplicated with the BACKUP command can only be used by the RESTORE command to restore a fixed disk drive's files. Do not use BACKUP files for archival purposes.

### Options

The following options may be used with the BACKUP command:

- /S**     Back up subdirectories also.
- /M**     Only back up those files that have changed since the last backup.

- /A** Add the files to be backed up to those already on the backup diskette. This does not erase old files on the diskette.
- /P** Pack as many files as possible onto each diskette. This creates a subdirectory on the diskette if it is the only way to fill the diskette.

This option may cause compatibility difficulties between files used on computers from different vendors.

- /D** Only back up those files that were last modified at or after a certain date.
- /T** Only back up those files that were last modified at or after a certain time.
- /L** Make a backup log entry in the file specified. If you don't give a file name for this log file, the default name BACKUP.LOG is entered in the root directory of the files being backed up.

## BACKUP

---

The first line of the entry in this file is: [date time], the backup dates and times. Each subsequent line in the file entry contains the backed up file name and the number of the diskette that contains the file. This information is used when you need to restore a specific file from a diskette.

If a BACKUP log file already exists, the current entries are appended to that file. This way you will always know which diskette to use for a RESTORE command.

---

**Purpose** The BREAK command is an MS-DOS toggle that sets the CONTROL-C check.

**Syntax** BREAK [ON|OFF]

**Comments** You use this toggle option to prevent MS-DOS from being affected by a **CTRL C** entry when the program you are using uses these keystrokes for another purpose.

Select BREAK OFF to turn off CONTROL-C when you are running a program that uses these keystrokes. Select BREAK ON when you return to MS-DOS.

If you do not select ON or OFF, MS-DOS displays the current setting of BREAK.

Set BREAK ON to stop assemble or compile processes. See the *System Programmer's Guide* for more information.

# CHDIR

## (CHange DIRectory)

---

**Purpose** The CHDIR command changes the current directory to another path or it displays the current subdirectory you are using.

**Synonym** CD

**Syntax** CHDIR [pathname]

**Comments** If your working directory is \BIN\USER\JOE and you want to change your path to another directory (such as \BIN\USER\JOE\FORMS), type:

**CHDIR \BIN\USER\JOE\FORMS**

and MS-DOS puts you in the new directory.

There is a simpler notation you can use with this command:

**CHDIR ..**

This command puts you in the parent directory of your working directory.

If you enter CHDIR without a pathname, your working directory is displayed. For example, if your working directory is \BIN\USER\JOE on drive B:, and you type

**CHDIR**

and press the **RETURN** key, the screen displays:

**B: \BIN\USER\JOE**

CHDIR is useful if you forget the name of your working directory.

To return to your root directory, type either

**CHDIR\**

or

**CD\**

# CHKDSK

## (CHecK DiSK)

---

<b>Purpose</b>	The CHKDSK command scans the directory of a specified disk to check that it is consistent with the files on the disk.
<b>Syntax</b>	CHKDSK [d:] <filespec> [/F] [/V]
<b>Comments</b>	It is a good idea to use CHKDSK from time to time to check for errors in a diskette or fixed disk directory. If errors are found, CHKDSK may display error messages.

CHKDSK displays a status report.

This is a sample status report:

```
160256 bytes total disk space
 8192 bytes in 2 hidden files
 512 bytes in 2 directories
30720 bytes in 8 user files
121344 bytes available on disk

65536 bytes total memory
53152 bytes free
```

---

**Options**

The following options may be used with the CHKDSK command:

**/F**      Fix any errors found in the directory.

If you use the /F option, these errors are automatically corrected:

- Invalid drive specification
- Invalid parameter
- Invalid subdirectory entry
- Cannot CHDIR to root  
Processing cannot continue
- First cluster number is invalid  
entry truncated
- Allocation error, size adjusted
- Has invalid cluster, file truncated
- Disk error reading FAT
- Disk error writing FAT
- <filename> contains non-contiguous  
blocks

**/V** Display messages on the screen while CHKDSK operates.

You can redirect the output from CHKDSK to a file. Simply type:

**CHKDSK A:>filename**

The errors are sent to the filename specified. Don't use the /F switch if you send the CHKDSK error messages to a file.

---

<b>Purpose</b>	The CLS command clears the display screen.
<b>Syntax</b>	CLS
<b>Comments</b>	<p>The CLS command is often used in batch processing files to clear the display before another command is begun.</p> <p>The CLS command causes MS-DOS to send the ANSI escape sequence ESC[2J (to clear your screen) to your display.</p>

# COMP

---

<b>Purpose</b>	The COMP command compares the contents of a file or group of files with others to verify that they are identical.
<b>Syntax</b>	COMP [<pathname1> [<pathname2>]]
<b>Comments</b>	<p>&lt;pathname1&gt; is the file or group of files that are to be compared with those specified in &lt;pathname2&gt;.</p> <p>More than one file may be specified if you use wild card characters. Only files that have matching names to those specified in &lt;pathname1&gt; are compared.</p> <p>Files may be on either the same or different drives. Files may be in the same or different directory.</p>

---

If the files to be compared are identical, the following message appears on your screen:

Files compare OK  
Compare more files (Y/N)?

Press **N** to stop the COMP command.

Press **Y** to repeat the procedure with another set of files.

If more than a single file is being compared, the COMP process continues until all files have been compared.

The COMP command displays an error message if one of these problems occurs:

- a specified directory path is invalid
- two files to be compared are of a different size
- a file specified by pathname2 cannot be found.

An advisory message is displayed if in the same location of two compared files the data does not match. The message indicates the offset in bytes (within the files) and the contents of those sections of the files.

If ten data mismatches are found, the COMP command halts and displays this message:

10 mismatches - ending compare

If an End-Of-File marker is not found at the end of a COMP command process, this message appears on your screen:

Eof mark not found

Some applications programs use files that are recorded in exact 128 byte multiples. Sometimes the actual data in these files may be less than exactly 128 bytes and the COMP program finds mismatches at the end of these types of files.

If the EOF message appears on your screen, it does not necessarily mean that there is a problem. It may be telling you that compare errors may not have occurred in the usable data portion of that file.

---

**Purpose** The COPY command copies one or more files to another disk.

The COPY command also lets you add the contents of one or more files to the end of another file. This is called concatenating (joining) files.

**Syntax** **To COPY a file:**

```
COPY [<pathname>] [<pathname>] [/V] [/A]  
[/B]
```

**To CONCATENATE a file:**

```
COPY <filespec> + <filespec> ... <filespec>
```

**Comments** **To COPY files:**

If the source and destination files are in the working directory, you do not need to specify a complete pathname.

If the second pathname option is not given, the copy is sent to the default drive using the original file name (first pathname option). If the first pathname is on the default drive and you don't specify the second pathname, COPY halts. You may not copy a file to itself.

## COPY

---

MS-DOS displays the error message:

```
File cannot be copied onto itself  
0 File(s) copied
```

If the second pathname entry is a drive designation, the file is copied using the original filename.

If the second pathname entry is a filename without a drive specifier, the original file is copied on the default drive with a different filename.

If the second pathname option is a complete file specification, the file is copied to the destination drive using the specified filename.

### Options

**/V** MS-DOS verifies that the file is being correctly copied on destination disk.

When you use the /V option, the COPY command runs more slowly because of the verification process.

You would use this option whenever you want to be certain that important data is being correctly recorded.

---

**/A** Indicates that the file is an ASCII file. This option entry applies to the file specification it precedes and to all subsequent file specifications on the command line until a **/B** is encountered.

In a source file specification **/A** allows data to be copied up to, but not including an End-Of-File marker. (In EDLIN, this is a **CTRL Z**. No more data in the file is copied after a **CTRL-Z** is encountered.

In a destination file specification **/A** adds an End-Of-File character to the end of the copied file.

**/B** Indicates that the file is a binary data file. This option refers to the preceding file specification and remains in effect until **/A** is encountered.

In a source file specification **/B** copies the entire file, including an End-Of-File marker.

In a destination file specification **/B** causes no End-Of-File marker to be added to the copied file.

### To CONCATENATE files:

The COPY command lets you join files (concatenation) while copying. This is done by simply listing any number of files as options in the COPY command entry, each separated by a plus sign (+).

For example,

**COPY MON.TX + TUE.TX APR22.TX**

joins the files MON.TX and TUE.TX, copying them into the file APR22.TX on the default drive.

You can also use wild card characters to combine several files into a single file.

For example,

**COPY \*.LST COMBIN.PRN**

takes all of the files with the extension .LST and combines them into the file named COMBIN.PRN.

---

You can use wild card characters for more complicated file joinings. For example, the COPY command entry

**COPY \*.LST + \*.REF \*.PRN**

joins each file with the extension .LST with a **corresponding** file with the extension .REF. A file named FILE1.LST is combined with a file named FILE1.REF (and so on) into a file named FILE1.PRN; a file named ABC.LST is combined with a file named ABC.REF to make a file named ABC.PRN.

This same technique can be used to combine all of the files with the extension .LST and the extension .REF into a single file by entering this command:

**COPY \*.LST + \*.REF COMBIN.PRN**

The resulting file is named COMBIN.PRN.

Be careful not to enter a concatenate COPY command where the destination file has the same extension as one of the source files. For example,

**COPY \*.REF ALL.REF**

would result in an undetected error if ALL.REF already existed on the destination disk. At the point the error would be detected, the ALL.REF file would have been destroyed.

COPY compares the source filename with the destination filename. If they are the same, that single source file is skipped over. This error message is displayed:

Content of destination lost before copy  
and the file joining process continues normally.

You can also “sum” files. For example,

**COPY ALL.LST + \*.LST**

appends all \*.LST files, except ALL.LST itself, to ALL.LST. This command entry is the correct way to append files with the COPY command. No error message is displayed.

---

<b>Purpose</b>	The CTTY command lets you change the device used by MS-DOS to accept command input. ("TTY" is what MS-DOS calls your keyboard.)
<b>Syntax</b>	CTTY <device>
<b>Comments</b>	The <device> is the device from which commands are to be sent to MS-DOS. You use this command if you want to change the device where you are working.

For example, the entry

## **CTTY AUX**

moves all command I/O (input/output) from the display keyboard to the AUX device such as a printer. The command

## **CTTY CON**

returns command I/O back to the display keyboard. The valid legal device names are described in Chapter 2, **Reserved Filenames and Extensions**.

# DATE

---

- Purpose** The DATE command is used to enter or change the date used by MS-DOS to note in a directory when a file was recorded.
- Syntax** DATE [<mm>-<dd>-<yy>]
- Comments** If you type DATE, this message appears:
- Current date is <mm>-<dd>-<yy>  
Enter new date: \_
- Press the **RETURN** key to leave the displayed date as shown.
- If you type DATE followed by a date entry, the new date is used by MS-DOS. For example,
- DATE 3-9-81**
- makes your computer use the date March 9, 1981.

---

A DATE entry must be entered using numerals only; letters are not permitted. The allowed options are:

<mm> = 1-12  
<dd> = 1-31  
<yy> = 80-99 or 1980-2099

You can use hypens (-) or slashes (/) to separate the date, month, and year entries. MS-DOS handles months and years correctly, whether the month has 31, 30, 29, or 28 days, including leap years.

**Note**

You can change the date from within a batch processing file.

# DElete

---

Purpose	DElete is the same as ERASE. This command removes designated files from a directory.
Synonym	ERASE
Syntax	DEL [pathname]
Comments	<p>Be careful. If you use wild card characters to describe a pathname, you may DElete more files than you intended. For example, the pathname</p> <p><b>DEL A:*. *</b></p> <p>causes the message</p> <p>Are you sure?</p> <p>to be displayed. If you type Y or y as a response, then <b>all</b> the files on that diskette are removed.</p> <p>You can use ERASE instead of DElete.</p>

---

<b>Purpose</b>	The DIR command displays all of the files in a specified directory.
<b>Syntax</b>	DIR [pathname][/P][/W]
<b>Comments</b>	<p>All the directory entries on the default drive are displayed if you only enter DIR.</p> <p>If you enter a drive specification (DIR d:), all directory entries from the specified drive are displayed.</p> <p>If you enter a filename without an extension (DIR filename), then all of the files in the directory with that filename are displayed.</p> <p>If you enter a complete file specification (DIR d:filename.ext), each file in the directory that has that exact file specification is displayed.</p> <p>Files are always displayed with their size in bytes, the time, and the date they were last modified.</p> <p>You can use wild card characters with the</p>

DIR command. See the chart below.

Command:	Equivalent
DIR	DIR *.*
DIR FILENAME	DIR FILENAME.*
DIR .EXT	DIR *.EXT

Options

There are two options with the DIR command.

- /P** selects Page Mode. With the /P option the directory pauses after the screen is filled (A “page” is displayed.) To restart the display of the directory, press any key.
- /W** selects Wide Display. With the /W option only the filenames are displayed, five filenames per line. No other file information is displayed.

---

**Purpose** Compares the contents of the diskette in the first specified drive to the contents of the diskette in the second specified drive. Usually, you would run DISKCOMP after a DISKCOPY operation to ensure that the two diskettes are identical.

This command is used only for comparing diskettes. If a fixed disk drive letter is specified, an error message is displayed.

This command compares two *entire diskettes*; the COMP command compares two *files*

**Syntax** DISKCOMP[d:] [d:] [/1] [/8]

**Remarks** You can specify the same drive or different drives in this command. If you specify the same drive, a single-drive comparison is performed. You are prompted to insert the diskettes at the appropriate time. DISKCOMP waits for you to press any key before it continues.

DISKCOMP compares all tracks on a track-for-track basis and issues a message if the tracks are not equal. The message indicates the track number and the side (0 or 1) where the mismatch was found.

After completing the comparison, DISKCOMP prompts:

Compare more diskettes (Y/N)?\_\_

To end the command, press N.

If you omit the second parameter, the default drive is used as the secondary drive. If you specify the default drive in the first parameter, this also results in a single-drive comparison.

DISKCOMP automatically determines the number of sides and sectors per track to be compared, based on the diskette that is to be read first (the first drive parameter entered).

### Options

- /L**      The /l parameter forces DISKCOMP to compare only the first side of the diskettes, even if the diskettes and drives are dual-sided.
- /8**      The /8 parameter causes DISKCOMP to compare only 8 sectors per track, even if the first diskette contains 9 sectors per track.

---

**Purpose** The DISKCOPY command copies all of the files on the source disk to a destination disk. The diskettes must be in separate drives unless you have a single diskette drive computer.

**Syntax** DISKCOPY [d:] [d:]

**Comments** The first modifier is the source drive, the second is the the destination drive.

**WITH A SINGLE DRIVE COMPUTER:** You are prompted to insert diskettes at the appropriate times. DISKCOPY pauses between insertions until you press any key before continuing.

After copying, DISKCOPY prompts:

Copy complete  
Copy another (Y/N)?\_\_

To copy another diskette, press Y.

To end the COPY, press N.

## DISKCOPY

---

### Notes

1. If you omit both options, a single-drive copy operation is performed on the default drive.
2. If you omit the second option, the default drive is used as the destination drive.
3. Both disks must have the same number of physical sectors and those sectors must be the same size.
4. Disks that have had a lot of file activity (add and delete) are fragmented, because space on the disk is no longer sequential. The first free sector found is the next sector allocated regardless of its location.

A fragmented disk performs slowly due to delays from finding, reading, or writing a file. Use the COPY command to copy a diskette that has had a lot of activity.

For example:

**COPY A:.\* B:**

copies all files from the disk in drive A: to the disk in drive B: sequentially, eliminating the fragmentation of space.

5. DISKCOPY automatically determines the number of sides to copy, based on the source drive and disk.

---

<b>Purpose</b>	The EXE2BIN command converts .EXE (executable) files to binary file format. Using EXE2BIN results in faster program loading and conserves space on the disk.
<b>Syntax</b>	EXE2BIN <filespec> [d:][<filename>[<.ext>]]
<b>Comments</b>	Refer to the MS-DOS <i>System Programmer's Guide</i> for more information.

# EXIT

---

**Purpose** The EXIT command lets you leave MS-DOS and return to the application program you are running on your computer. Where your program permits, you must enter **COMMAND.COM** first.

**Syntax** EXIT

**Comments** EXIT is used when you are in an application program and need to use the MS-DOS command processor (COMMAND.COM), then return to your program.

For example, to examine a directory on drive B: when you are running a program, you type **COMMAND** at the **A>** prompt, then use the **DIR B:** command to look at the directory of the files on drive B:.

```
A>COMMAND  
A>DIR B:
```

(listing appears)

To return to your program you enter the EXIT command.

---

**Purpose**            The FDISK command lets you set up a fixed disk drive to receive multiple operating systems.

**Syntax**            FDISK

**Comments**        FDISK lets you create and define the operating system partitions on your fixed disk.

Refer to Chapter 3, **Using a Fixed Disk**, for complete information.

## (F)ile (C)ompare

---

<b>Purpose</b>	The File Compare utility compares the contents of two files.
<b>Syntax</b>	<b>FC [/#][/B][/C][/W] filename1 filename2</b>
<b>Comments</b>	<p>If you have copied a file and modified the copy, you may later want to compare copies to check the changes you have made. You can use File Compare to do this and direct that the differences between the two files be output to the screen or to a third file.</p> <p>The files being compared may be either source files (files containing source statements of a programming language) or binary files (output from the MACRO-86 assembler, the MS-LINK linker utility, or a high-level language compiler).</p> <p>The comparisons are either on a line-by-line or a byte-by-byte basis. Line-by-line comparison isolates blocks of lines that are different between the two files and prints those blocks of lines. This is the default method. If there are too many differences (too many lines), the program simply reports that the files are different and stops. If no matches are found after the first difference, FC displays:</p> <p>*** Files are different ***</p> <p>and returns to MS-DOS default drive prompt.</p>

The byte-by-byte comparison displays the bytes that are different between the two files. For binary files larger than available memory, FC compares both files completely, overlaying the portion in memory with the next portion from disk. All differences are output in the same manner as those files that fit completely in memory.

## Options

The following options are used with FC.

- /#**      Number of lines required to match for the files to be considered as matching again after a difference has been found. # can be any number from 1 to 9. Default is 3. Use only in source comparisons.
- /B**      Binary comparison of both files is performed. The two files are compared byte-by-byte with no attempt to re-synchronize after a mismatch. Mismatches are printed as follows:

```
--ADDRS----F1----F2-  
xxxxxxx yy  zz
```

where xxxxxx is the relative address of the pair of bytes from the beginning of the file. Addresses start at 00000000; yy and zz are the mismatched bytes from file1 and file2 respectively.

If one of the files contains less data than the other, a message appears. If, for example, F1 ends before F2:

\*\*\*Data left in F2\*\*\*

- /C Ignore case of letters. All letters in the files are considered upper case. Use only in source comparisons.
- /W Compress whitespace (tabs and spaces) during comparison. Multiple contiguous whitespace in any line is considered as a single white space. Only beginning and ending whites are ignored. Use only in source comparisons.

**Examples**

Assume these two ASCII files are on disk:

ALPHA.ASM

BETA.ASM

A  
B  
C  
D  
E  
F  
G  
H  
I  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
W  
X  
Y  
Z

A  
B  
C  
G  
H  
I  
J  
1  
2  
P  
Q  
R  
S  
T  
U  
V  
4  
5  
W  
X  
Y  
Z

One way to compare them is:

FC ALPHA.ASM BETA.ASM<cr>

FC compares the two files and displays the differences on the screen. All defaults remain intact. The output appears as follows.

(F)ile (C)ompare

---

\_\_\_\_\_ALPHA.ASM  
D  
E  
F  
G

NOTE: ALPHA file contains  
DEFG, BETA contains G

\_\_\_\_\_BETA.ASM  
G

\_\_\_\_\_ALPHA.ASM  
M  
N  
O  
P

NOTE: ALPHA contains MNO  
where BETA contains J12

\_\_\_\_\_BETA.ASM  
J  
1  
2  
P

\_\_\_\_\_ALPHA.ASM  
W

NOTE: ALPHA contains W  
where BETA contains 45W

\_\_\_\_\_BETA.ASM  
4  
5  
W

---

If you use:

FC /B ALPHA.ASM BETA.ASM<cr>

The following binary comparison appears.

—ADDRS—	F1—	F2—
00000009	44	47
0000000C	45	48
0000000F	46	49
00000012	47	4A
00000015	48	31
00000018	49	32
0000001B	4D	50
0000001E	4E	51
00000021	4F	52
00000024	50	53
00000027	51	54
0000002A	52	55
0000002D	53	56
00000030	54	34
00000033	55	35

\*\*\* DATA left in F1 \*\*\*

# FIND

---

<b>Purpose</b>	The FIND command searches for a specified string of text in a file or files.
<b>Syntax</b>	FIND [/V /C /N] <"string"> [<filename...>]
<b>Comments</b>	<p>FIND is an MS-DOS filter. A filter discards all characters or values except those that match its selection criteria.</p> <p>FIND accepts entries that are the object of the search (the text string) and the place or places (the file or files) where it is to look for matching data. The term for one of these entries is "argument."</p> <p>FIND displays all lines that contain the data being sought from the files specified in the command line.</p> <p>If no files are specified, FIND takes the input on the screen and displays all lines that contain the specified string.</p>

---

<b>Options</b>	<b>/V</b>	causes FIND to display all lines not containing the specified string.
	<b>/C</b>	causes FIND to print only the count of lines that contained a match in each of the files.
	<b>/N</b>	causes each line to be preceded by its relative line number in the file.

The text string must be enclosed in quotes. If a text string you are looking for is already enclosed in quotes, the entry on the command line must be in double quotes.

## Examples

**FIND "Fool's Paradise" BOOK1.TXT BOOK2.TXT**

displays all lines from BOOK1.TXT and BOOK2.TXT (in that order) that contain the string "Fool's Paradise." The command

**DIR B: | FIND /V "DAT"**

causes MS-DOS to display all names of the files on the disk in drive B: that do not contain the string DAT.

# FORMAT

---

<b>Purpose</b>	FORMAT prepares diskettes or a fixed disk to receive data from your computer. Diskettes or a fixed disk must be formatted before they can be used.
<b>Syntax</b>	FORMAT [d:][/O][/V][/S]
<b>Comments</b>	FORMAT initializes the directory and file allocation tables on the specified drive. If no drive is specified, the diskette in the default drive is formatted.
<b>Options</b>	<p><b>/O</b> causes FORMAT to produce a PC DOS version 1.X compatible disk. The /O option causes FORMAT to reconfigure the directory with an OE5 hex byte at the start of each entry so that the disk may be used with 1.X versions of PC DOS, as well as MS-DOS 1.25/2.00 and PC DOS 2.00.</p> <p>This option should only be given when needed because it takes a fair amount of time for FORMAT to perform the conversion, and it noticeably decreases 1.25 and 2.00 performance on disks with few directory entries.</p>

- /V** causes FORMAT to prompt for a volume label after the disk is formatted.
- /S** If this option is used, it must be the last entry on the command line. /S copies the hidden SYS files onto the diskette or fixed disk.

The files are copied in the following order:

- IO.SYS
- MSDOS.SYS
- COMMAND.COM

# GRAPHICS

---

<b>Purpose</b>	The GRAPHICS command lets you print any graphics displayed on the screen if you have a graphics-compatible printer attached to your computer.
<b>Syntax</b>	GRAPHICS
<b>Comments</b>	<p>After loading GRAPHICS into the memory of your computer, press and hold <b>SHIFT</b>, then press <b>PR SCR</b> to direct the graphics displayed on the screen to the graphics-compatible printer.</p> <p>The GRAPHICS command defaults to a standard printer. Use the MODE command to set other parameters that are required for your own printer.</p> <p>If color graphics are used, the screen image is printed in up to 4 shades of gray.</p> <p>Use interrupt #5 (INT5) to print the screen from a program. Refer to the <i>System Programmer's Guide</i> for more information.</p>

---

<b>Purpose</b>	The MKDIR (MaKe DIRectory) command creates a new directory on the specified pathname.
<b>Syntax</b>	MKDIR <pathname>
<b>Synonym</b>	MD
<b>Comments</b>	You use MKDIR to create a hierarchical directory structure on your fixed disk. You create subdirectories from your root (top) directory by using the MKDIR command.

For example, the command entry

**MKDIR \USER**

creates a subdirectory \USER in your root directory. To create a directory named JOE under \USER, you enter:

**MKDIR \USER \JOE**

Refer to Chapter 3, **Using a Fixed Disk**, for more information about different ways to use directories and subdirectories to your best advantage.

# MODE

---

**Purpose** The MODE command lets you define some of the operating characteristics of your computer. There are three different uses of MODE.

**Syntax** To set communication protocol for the RS-232C serial interface port:

MODE COMn:baud[,parity[,databits[,stopbits[,p]]]]

To set the type of display you are using and how many characters are displayed on a single line:

MODE n[[[,R|L],T]

To set printer parameters or to redirect printer output to the RS-232C serial interface port:

MODE LPT#: [chars][,spacing]

or

MODE LPT# = COMn

---

**Comments****Setting Communication Protocols**

```
MODE COMn:baud[,parity[,databits[,stopbits[,p]]]]
```

When you are setting the communication protocol for your serial interface port you must specify which port you are setting and the speed (baud rate) you want to use. Parity, databits, and stopbits entries are optional. You may also specify that the serial port continuously attempts to make and maintain contact with a printer or telephone coupler following time-out errors.

The syntax element **n** must be entered as either **1** or **2**. **1** is the built-in serial interface port. **2** is an optional second serial interface port.

The syntax element **baud** is the baud rate that the serial port uses to communicate. Only the first two digits of the baud rate need to be entered. The baud rates are: 110, 300, 600, 1200, 2400, 4800, or 9600.

Parity setting is optional. The entry may be either **e** (even parity), **o** (odd parity), or **n** (no parity). The default value if no entry is made is even parity.

## MODE

---

Data bits are an optional entry. If no entry is made, the default value is 7. You may enter either 7 or 8 for the number of data bits in a character.

Stopbits are an optional entry. You may enter either 1 or 2 for the number of stopbits at the end of a character. If no stopbits entry is made, the default value is 2 stopbits if the baud rate is 110. The default value for all other baud rates is 1 stopbit.

The **p** entry in the command line is optional. If it is included, your serial interface port will continuously retry to send if time-out errors occur. Some types of printers may not match the timing requirements of your serial port. In most cases, the **p** option overcomes this problem.

### Examples

MODE COM1:96

The form of the MODE command sets the baud rate to 9600 for the built-in serial interface port. All other entries are set to their default values.

MODE COM1:1200,N,1,P

This form of the MODE command sets the built-in serial port to 1200 baud, no parity, one stop bit, and continuous retry on time-outs. This command format might be used with a high-speed telephone coupler.

## Comments

Setting Display Characteristics

MODE n[[,R|L],T]

Depending upon the type of display you are using with your computer, or if you have more than one display attached to your computer, you use this form of the MODE command to set the display type and format.

This form of the command requires an entry called an argument. Arguments are values that tell the computer what option you want.

## Options

- |    |  |
|----|--|
| 40 | Sets the width of the display line to 40 characters. This is used with a color display only. |
| 80 | Sets the width of the display line to 80 characters. This is used with a color display only. |

## MODE

---

- BW40** If you have a monochrome and a color display attached to your computer, this selects the color monitor to be active, sets the display mode to black and white and to 40 characters per line.
- BW80** Same as above, only 80 characters per line.
- CO40** Same as BW40, only sets the display mode to color and to 40 characters per line.
- CO80** Same as BW80, but color mode is set.
- MONO** If you have a monochrome and a color display attached to your computer, this selects the monochrome display to be active. The display width for monochrome displays is always 80 characters per line.
- R** Shifts display one character to right.
- L** Shifts display one character to left
- T** displays a test pattern and queries the user if alignment is correct

### Comments

### Setting the Printer Characteristics and Selecting the Printer Port

MODE LPT#: (chars)(spacing)

---

MODE LPT# = COMn

LPT# is the device name for the parallel printer port(s) in your computer. The # is the printer port number you enter to select which printer port you are setting. The number you enter may be either 1, 2, or 3. Port #1 is the built-in parallel port in your computer and is the default port for all printing activity.

The first example above lets you set the number of characters per inch that your printer will print and the vertical spacing between lines of output. For example,

MODE LPT1: 12,6

sets parallel port #1, then passes the instructions to your printer to print 12 characters per inch and 6 vertical lines per inch.

Some printers do not accept this type of command. Check the user guide that came with your printer for more information.

In the second example above, you reassign the printer output port to be a serial interface port in your computer. You may want to do this

## MODE

---

if you have a special printer attached to your computer that you use for correspondence-quality output or for graphics.

For example,

```
MODE LPT1 = COM2
```

redirects all printer output to an optional second serial interface port attached to your computer. COM1 is the name of the built-in serial port in your computer. COM2 would be an extra interface port that you may have added at a later time.

### Note

In some cases you may need two MODE commands to set up the serial interface port for use with a printer. For example,

```
MODE LPT1: 2400,E,1,P  
MODE LPT1 = COM1
```

would be required to set the printer port baud rate to 2400, even parity, 1 stop bit and continuous retry. The second MODE command redirects the output using the new settings to the built-in serial port in your computer.

---

**Purpose** The MORE command limits the amount of information displayed on the screen to a single page at a time.

**Syntax** MORE

**Comments** The MORE command is an MS-DOS filter. A filter is a special program that takes the output from another command and modifies it.

The MORE command takes input (such as a command from your keyboard) and displays one screen of information at a time. — MORE — is displayed at the bottom of your screen and the display pauses until you press **RETURN**.

This process continues until all of the input data has been read from the source file.

In the example below the vertical bar (|) is used to pipe the output from the text file to the MORE filter.

## MORE

---

You use the MORE command to view a long file one screen at a time. For example

**TYPE MYFILES.COM | MORE**

displays the file MYFILES.COM (on the default drive) one screen at a time.

The MORE command can also be used with the output redirection character (>) to create a text file from the default device CON, your keyboard.

For example, the command entry

**more> <filename>**

opens the specified file on the default drive and lets you enter text from the keyboard. Pressing **CTRL C** ends the text entry and writes the text you've entered into the file.

---

<b>Purpose</b>	The PATH command sets a search path for all MS-DOS file commands.
<b>Syntax</b>	PATH [<pathname>[;<pathname>]...]
<b>Comments</b>	The PATH command lets you define where MS-DOS searches for external commands after it searches your working directory.

The default value is no path.

To tell MS-DOS to search your \BIN\USER\JOE directory for external commands, type:

**PATH \BIN\USER\JOE**

Every time you ask for an external MS-DOS command, MS-DOS searches \BIN\USER\JOE directory for external commands until you set another path or finish the current MS-DOS session.

MS-DOS will search more than one path if you separate the pathnames with semicolons. For example,

## PATH

---

**PATH \BIN \USER \JOE; \BIN \USER \SUE; \BIN \DEV**

tells MS-DOS to search the \JOE subdirectory first, then the \SUE subdirectory, and last to search the \BIN\DEV\ subdirectory for the external commands.

MS-DOS searches the pathnames in the order specified in the PATH command.

The command PATH with no options displays the current path.

If you use a semicolon to specify PATH, MS-DOS searches only the working directory for the external commands.

---

<b>Purpose</b>	The PRINT command lets you print a text file on your printer while you are working with other MS-DOS commands. This is called “background printing” which means that the printing is being done in the background while the other work you are doing with MS-DOS is going on in the foreground.
<b>Syntax</b>	PRINT [[filespec][[/T][/C][/P]]...
<b>Comments</b>	Use the PRINT command only if you have a line printer attached to your computer.
<b>Options</b>	<p>The following options are used with the PRINT command.</p> <p><b>/T</b>      TERMINATE stops the printing session and removes all files from the print queue that have not yet been printed.</p> <p><b>/C</b>      CANCEL suspends printing of the specified file and all subsequent files until a /P option is entered.</p> <p><b>/P</b>      PRINT begins printing. The preceding file and all subsequent files are added to the print queue until you enter a /C option.</p>

## PRINT

---

PRINT with no options displays the contents of the print queue on your screen without affecting the queue.

### Examples

```
PRINT /T
```

empties the print queue.

```
PRINT A:TEMP1.TST /C A:TEMP2.TST A:TEMP3.TST
```

the /C cancels the printed output of the TEMP1.TST file and suspends the operation of the print queue until the /P is entered.

```
PRINT TEMP1.TST /C TEMP2.TST /P TEMP3.TST
```

/C removes TEMP1.TST from the queue, /P adds TEMP2.TST and TEMP3.TST to the queue.

---

<b>Purpose</b>	The PROMPT command lets you assign the characters or phrase used as the MS-DOS prompt.
<b>Syntax</b>	PROMPT [<prompt-text>]
<b>Comments</b>	<p>The PROMPT command lets you set the MS-DOS system prompt (A&gt;) to any other text you want.</p> <p>If you don't enter any text with the PROMPT command, the prompt is reset to its default value, A&gt;.</p> <p>You can set the prompt to a special prompt, such as the current time, by using the characters indicated below.</p> <p>The following characters can be used in the prompt command to specify special prompts. They must all be preceded by a dollar sign (\$) in the prompt command:</p>

# PROMPT

---

Specify This Character	To Get This Prompt:
\$	The '\$' character
t	The current time
d	The current date
p	The current directory of the default drive
v	The version number
n	The default drive
g	The '>' character
l	The '<' character
b	The ' ' character
-	A CR LF sequence
s	A space (leading only)
h	A backspace
e	ASCII code X'1B' (escape)

## Examples

PROMPT \$n

Sets the default drive letter prompt.

PROMPT Time = \$t\$Date = \$d

Sets a two-line prompt that prints:

Time = (current time)

Date = (current date)

ANSI Escape sequences used by programmers can be used in the MS-DOS prompt. For example,

```
PROMPT $e[7m$n:$e]m
```

Sets the prompts in inverse video mode yet keeps text in normal video mode.

# RECOVER

---

**Purpose** The RECOVER command lets you recover a file or an entire diskette or fixed disk drive that has bad sectors.

**Syntax** RECOVER <filename | d:>

**Comments** If a sector on a disk is bad, you can recover either the file containing that sector (without the bad sector) or the entire diskette or fixed disk drive (if the bad sector was in the directory).

To recover a particular file, you enter

**RECOVER <filename>**

causing MS-DOS to read the file sector by sector and to skip the bad sector(s). If MS-DOS finds bad sector(s), the sector(s) are marked so that MS-DOS no longer puts any data in that sector.

To recover a diskette or fixed disk, enter

**RECOVER <d:>**

where d: is the letter of the drive containing the disk to be recovered.

The vertical bar (|) pipes the output from the file to a specified drive.

---

<b>Purpose</b>	The REM command lets you see any text that is on the same line as the REM command when a batch processing file is being run.
<b>Syntax</b>	REM [comment]
<b>Comments</b>	Use only the space, tab, and comma characters in the text of a REM comment. Other punctuation may be interpreted as a drive specifier or part of an MS-DOS command.

**Example:**

```
REM This file checks new disks
REM It is named NEWDISK.BAT
PAUSE Insert new disk in drive B:
FORMAT B:/S
DIR B:
CHKDSK B:
```

# REN

---

<b>Purpose</b>	The RENAME command lets you change a file name to another file name.
<b>Synonym</b>	RENAME
<b>Syntax</b>	REN <pathname> <filename>
<b>Comments</b>	The first modifier <pathname> is the name of the file you want to change. The second modifier <filename> is the new filename you want to give to the file.

The first file specification must include a drive designation if the file is on another drive than the default drive. The new file name is always created on the same drive or pathname as the source file.

You can use wild card characters in either option. All files matching the first file specification are renamed. If wild card characters appear in the second filename, corresponding character positions are not changed.

For example, this command changes the names of all files with the .LST extension to similar names with the .PRN extension:

**REN \*.LST \*.PRN**

With this command, REN renames the file FLYER on drive B: to CRYER:

**REN B:FLYER CR???**

The file remains on drive B:.

**Purpose** The RESTORE command puts the files backed up using the BACKUP command back on the diskette or fixed disk drive from which they came.

**Syntax** RESTORE <d:>  
[<d:][<path>][<filespec>][/S][/P]  
[/B:<date>][/A:<date>][/E:<time>]  
[/L:<time>][/M][/N]

**Comments** The first modifier you specify is the drive designator of the disk containing the backed up files. The second parameter is the file specification indicating the files you want to restore.

The RESTORE command may not operate in the way you expect with files created with the BACKUP program supplied by other vendors.

**Options**

/S	Restore subdirectories also.
/P	If any hidden or read-only files match the file specification, prompt for permission to restore them.
	If omitted, all hidden and read-only files are copied.

## RESTORE

---

- /B** Only restore those files that were last modified on or before the given date.
- /A** Only restore those files that were last modified on or after the given date.
- /E** Only restore those files that were last modified at or earlier than the given time.
- /L** Only restore those files that were last modified at or later than the given time.
- /M** Only restore those files that have been modified since the last backup.
- /N** Only restore those files that no longer exist on the destination disk.

---

<b>Purpose</b>	The RMDIR command removes a subdirectory from a hierarchical directory structure.
<b>Synonym</b>	RD
<b>Syntax</b>	RMDIR <pathname>
<b>Comments</b>	<p>You use the RMDIR command to remove an <b>empty</b> directory. All the files must be first DELETED or ERASEd, except for the subdirectory shorthand symbols . and .. .</p> <p>The single period (.) means the current directory. Double periods (..) mean the parent directory.</p> <p>To remove the \BIN\USER\JOE directory, you first check the directory with a DIR and path-name to check that there aren't any important files that you want to keep, then you enter:</p> <p><b>RMDIR \BIN\USER\JOE</b></p> <p>The directory has been deleted from the directory structure.</p>

# SET

---

**Purpose** The SET command is used to set one string value to be equivalent to another string for use by later programs.

**Syntax** SET [<string = string>]

**Comments** The SET command is only used if you want to set values that will be used by programs you have written.

An application program checks the values that have been set with the SET command by issuing a SET with no options.

For example,

**SET TTY = VT100**

sets your TTY value to VT100 until you change it with another SET command.

More information about using the SET command appears in Chapter 6, **Batch Processing Commands**.

---

<b>Purpose</b>	The SORT command is used to sort the contents of a file and send the result to a device or file you select. The file sorted can also be a directory or information that would be displayed on your screen.
<b>Syntax</b>	<code>SORT [/R] [/+ n]</code>
<b>Comments</b>	<p>Use the symbols &lt; and &gt; to direct how SORT is to accept and send information. The symbol "&lt;" means "accept data from this file" and the symbol "&gt;" means "send the output data from the command to this file."</p> <p>You can use SORT to alphabetize a file by a specified column, such as one of the file statistic columns in a DIRectory listing.</p>
<b>Options</b>	<p>The following options are used with the SORT command.</p> <p><b>/R</b>      SORT is done in ASCII order unless you specify the /R option. This option reverses the sort; that is, sorts from Z to A.</p>

## SORT

---

**/+ n** SORT starts with column n where n is some number. If you do not specify this option, SORT begins sorting with the first column of input.

### Examples

The following command reads the file UNSORT.TXT, reverses the sort order, then writes the result to a file named SORT.TXT:

**SORT /R <UNSORT.TXT >SORT.TXT**

The following example sends (using the “pipe” symbol |) the DIR command listing to the SORT filter. The SORT command sorts the directory listing starting with column 14 (the column that contains file size in the DIR listing), then sends the result to the screen.

The result of this command is a directory sorted by file size:

**DIR | SORT /+ 14**

The command

**DIR | SORT /+ 14 | MORE**

does the same thing as the previous example, except that the MORE filter lets you read the sorted directory listing one screen at a time.

---

**Purpose**                The SYS command sends MS-DOS system files to the specified disk drive from the default drive.

**Syntax**                SYS <d>:

**Comments**            You use SYS to update MS-DOS or to place system files on a formatted disk that does not already have any files on it.

You must enter a drive designation (d:) with the SYS command.

Be careful. Diskettes that contain an earlier version of the MS-DOS SYS files are probably not the same size as the system files you transfer using the SYS command.

It is safer to use the /S option with the FORMAT command to place a current version of the SYS files on a new diskette, then COPY all data and program files over to the new diskette.

Your destination disk must be completely blank or already have the MS-DOS files IO.SYS

and MSDOS.SYS. The files are copied in the following order:

IO.SYS  
MSDOS.SYS

IO.SYS and MSDOS.SYS are both hidden files that don't appear when the DIR command is performed. COMMAND.COM (the command processor) is **not** transferred with a SYS command.

Use the COPY command to move the COMMAND.COM file to the destination diskette.

---

<b>Purpose</b>	The TIME command lets you display and set the time of day used by your computer to mark when files were last written or changed.
<b>Syntax</b>	TIME [<hh>[:<mm>]]
<b>Comments</b>	<p>If only the TIME command is entered without any values for the current time, your computer displays:</p> <pre>Current time is &lt;hh&gt;:&lt;mm&gt;:&lt;ss&gt;.&lt;cc&gt; Enter new time:.</pre> <p>If you don't want to change the time shown, press the <b>RETURN</b> key.</p> <p>You may enter a new value for the time of day at any time by simply typing the TIME command and a value. For example,</p> <pre><b>TIME 8:20</b></pre> <p>sets the new time of day to 8:20am.</p>

## TIME

---

The value for TIME is entered using numerals only; letters are not allowed. The valid values are:

<hh> = 00-24  
<mm> = 00-59

The hour and minute entries must be separated by colons. You cannot set the <ss> (seconds) or <cc> (hundredths of seconds) values.

MS-DOS uses the time entered as the new time if the values and separators are valid. If the values or separators are not valid, MS-DOS displays the message:

Invalid time  
Enter new time:..

Make a valid entry for the current time and press the **RETURN** key.

---

<b>Purpose</b>	The TREE command lets you view the directories and paths on a specified drive. TREE also lists the files in each directory.
<b>Syntax</b>	TREE [drivespec:][/F]
<b>Comments</b>	<p>The command modifier [drivespec:] is the drive that contains the diskette or fixed disk that contains the directories to be displayed.</p> <p>For each directory on the specified drive, TREE displays:</p> <ul style="list-style-type: none"><li>• the directory path starting from the root directory</li><li>• the subdirectories within the directory</li><li>• optionally, the files in each directory</li></ul>
<b>Options</b>	<p><b>/F</b>      The /F option displays the filenames and file specifications in each directory.</p>

# TYPE

---

<b>Purpose</b>	The TYPE command displays the contents of a file on the screen.
<b>Syntax</b>	TYPE <filespec>
<b>Comments</b>	<p>You use the TYPE command to check the contents of a file without modifying it or needing to use another program to display the contents of a file.</p> <p>The file that you process with a TYPE command should be an ASCII text file, otherwise the results of this command are unpredictable and usually result in a garbled display that does not give you enough information to determine what the file contains. Binary files that contain control characters will produce this type of garbled display.</p> <p>Use the MORE command with a TYPE command to view a screen page at a time.</p> <p>The only screen formatting command that the TYPE command recognizes and acts upon is TAB. TABs are expanded to eight spaces for each time a TAB is encountered in the file.</p>

---

<b>Purpose</b>	The VER command displays the revision level of MS-DOS that is in memory in your computer.
<b>Syntax</b>	VER
<b>Comments</b>	<p>If you are working with different revision levels of MS-DOS you use the VER command to determine which level of MS-DOS you are currently using.</p> <p>Some MS-DOS commands (DISKCOPY and DISKCOMP, for example) require that the diskettes be the same MS-DOS revision level in order to perform the task you request.</p>

# VERIFY

---

**Purpose** The VERIFY command is a toggle option that determines if a COPY command automatically confirms that a file has been correctly copied.

**Syntax** VERIFY [ON|OFF]

**Comments** The VERIFY command is the same as the /V option in the COPY command. If you want to verify that all of the files you copy are correctly written to the destination diskette or fixed disk drive, set the VERIFY command to an ON status.

When you set VERIFY to ON MS-DOS performs a VERIFY each time you use a COPY command. An error message is displayed if MS-DOS is unable to successfully write the file to the destination diskette or fixed disk.

If you want to know the current setting of VERIFY is, type VERIFY with no modifier.

---

<b>Purpose</b>	The VOL command displays the label, if any, of the specified diskette.
<b>Syntax</b>	VOL [d:]
<b>Comments</b>	<p>VOL [d:] displays the volume label of the disk in drive d:. If you don't specify [d:] MS-DOS displays the label of the diskette in the default drive.</p> <p>A diskette or fixed disk volume label is initially set with the /V option of the FORMAT command.</p>